



Artigo submetido ao Curso de Engenharia Civil da UNESC -  
como requisito parcial para obtenção do Título de Engenheiro Civil



# DESENVOLVIMENTO DE SOFTWARE PARA SIMULAÇÃO E SOLUÇÃO DE ESFORÇOS EM VIGAS COMO FERRAMENTA EDUCACIONAL

Marco Aurelio Viegas (1), Alexandre Vargas (2)

UNESC – Universidade do Extremo Sul Catarinense

(1) [marcoaurelioviegas@gmail.com](mailto:marcoaurelioviegas@gmail.com), (2) [avargas@uneschnet](mailto:avargas@uneschnet)

## RESUMO

Conhecendo as dificuldades para o entendimento dos assuntos estudados nas disciplinas de análise estrutural no curso de engenharia civil e, reconhecendo que a utilização de ferramentas computacionais pode ser de grande importância para o processo ensino-aprendizagem nessa área do conhecimento, buscou-se desenvolver um software que simulasse em tempo real as ações dos carregamentos em uma viga, apresentando como resultado as reações nos apoios, esforço cortante e momento fletor com seus respectivos diagramas. Em se tratando de um programa computacional, foi utilizado o método da rigidez direta, também conhecido como método dos deslocamentos na formulação matricial, que tem uma resolução que pode ser adequada de forma mais clara e eficiente na programação. O método consiste em encontrar os deslocamentos nodais e após isso as reações nos apoios. Contudo neste trabalho, as condições de carregamento criadas, tornam o esforço normal desprezível, por isso o objetivo está nas reações e nos diagramas de esforço cortante e momento fletor. Com uma interface didática, são infinitas as possibilidades de interação com o usuário para construir praticamente qualquer tipo de viga, com os mais diferentes tipos de carregamento e configurações de apoio, inclusive vigas hiperestáticas. Além disso, um dos diferenciais é que a cada interação do usuário com o software, os resultados são atualizados instantaneamente. Assim, modificar a posição de um apoio ou o valor de uma carga, possibilita ao usuário acompanhar e discutir os resultados obtidos.

*Palavras-chave: Software didático. Vigas. Esforços em vigas.*

## 1. INTRODUÇÃO

O uso de recursos visuais sempre foi um dos métodos mais eficazes de transmitir o conhecimento. Desenhos, gráficos e figuras sempre auxiliaram o entendimento e a disseminação da informação, desde a era primitiva, servindo como auto explicativa ou complementando ideias. Estudos mostram que quanto mais

sentidos do corpo humano são exigidos durante uma explicação, mais facilidade o cérebro tem para assimilar as informações. Talvez essa seja uma das explicações do porque dificilmente uma criança aprende a andar de bicicleta apenas olhando outra fazer.

Conhecendo a necessidade e dificuldade de entendimento de alguns assuntos, os professores buscam alternativas que facilitem a condução de suas disciplinas. Nesse século da informática, os recursos tecnológicos estão cada vez mais próximos do dia a dia das pessoas e inseridos também no processo ensino-aprendizagem. Historicamente os primeiros sistemas computacionais para uso no ensino surgiram por volta de 1960, inspirados no método de instrução programada, criado na década anterior, que consiste na organização do material a ser ensinado em seguimentos encadeados, denominados módulos. Esses módulos são apresentados ao estudante de forma gradual e sequencial. Assim, o aprendiz pode ditar seu ritmo, retornando aos módulos anteriores quando sentir necessidade ou avançar para módulos futuros caso tenha interesse (Valente, 1999). Entretanto esses sistemas computacionais não tiveram tanto sucesso e eficiência na prática, pois geralmente apresentavam ao final de cada módulo, perguntas que se não fossem respondidas conforme havia sido especificado no programa, impediam o estudante de avançar, havendo pouca interação entre homem e máquina.

Com o passar dos anos houve grande evolução nos computadores e por consequência nos sistemas educacionais que, em alguns casos, permitem "infinitas" interações com o usuário. Atualmente, nos ambientes de aprendizagem, existem diversos tipos de *softwares* educativos: para exercitação, tutoriais, simulações e até mesmo jogos. Em universidades e cursos técnicos é comum a inserção de *softwares* comerciais, utilizados pelos futuros profissionais da área, incluindo muitas vezes disciplinas específicas para ensino e utilização dessas ferramentas. Contudo, Lacerda (2007) defende que em geral, os *softwares* educativos estão sendo utilizados basicamente para fixação de um conteúdo trabalhado anteriormente em sala de aula ou simplesmente para a diversão ou entretenimento dos alunos. Devemos considerar que além do computador, aluno e *software*, é necessário também que o professor esteja capacitado e preparado para a utilização do computador/*software* como meio educacional, conforme destaca Valente (1993). O autor ainda afirma que o professor deve deixar de ser simplesmente o repassador de

conhecimento e começar a atuar como criador de ambientes de aprendizagem e facilitador do processo de desenvolvimento intelectual do aluno, utilizando o computador como aliado.

Com base nesses estudos e fundamentos, foi observada a necessidade de uma ferramenta que possibilitasse mais interações nas disciplinas que estudam esforços em vigas. Neste contexto, foi idealizado em um *software* onde o aluno ou professor pudessem inserir uma viga, os carregamentos e em tempo real os resultados fossem exibidos em tela, apresentando as reações nos apoios e os diagramas de esforço cortante e momento fletor simultaneamente. Com isso, cada modificação feita, seja ela na viga, nos apoios ou nos carregamentos, seria apresentada em tempo real nos diagramas.

Com o intuito de ser utilizado como um *software* didático foi definido que ele deve possuir uma interface simples, fácil e intuitiva, para que usuário possa em alguns minutos entender como funciona sem necessitar de treinamento, curso ou manual.

Aplicativos comerciais para cálculo de estruturas também podem oferecer recursos para demonstrar as reações nos apoios, os diagramas de esforço cortante e momento fletor, contudo, normalmente exigem algumas configurações e não apresentam os resultados em tempo real considerando a entrada e principalmente as possibilidades de alteração de vãos, cargas e apoios, dificultando o seu uso na educação.

Desta forma é esperado que o aplicativo desenvolvido nesse trabalho proporcione aos professores uma ferramenta dinâmica que irá colaborar para o desenvolvimento das aulas que envolvam as análises de vigas e, que por fim, facilite o entendimento, ampliando as interações, discussões e o interesse sobre esse assunto que é fundamental para formação de um engenheiro civil.

## 2. MATERIAIS E MÉTODOS

Este trabalho teve características teóricas onde foram realizadas revisões bibliográficas sobre conceitos de equilíbrio estático, força, momento, convenções de sinais e unidades, esforço cortante e momento fletor, além do método da rigidez direta. Em conjunto com essas revisões e formulações computacionais obtidas nos

estudos teóricos, o trabalho teve sua parte experimental através do desenvolvimento do software, simulando e comparando os resultados obtidos com diversas situações teóricas estudadas em sala de aula.

O aplicativo foi desenvolvido na linguagem de programação *Java*, que possui muitos recursos, alta segurança, estabilidade, independe da plataforma utilizada, além de possuir licença gratuita e ser amplamente difundida (ARNOLD et al: 2007). Para desenvolvimento da aplicação foi utilizado o ambiente de desenvolvimento *Oracle NetBeans* em um computador convencional com o sistema operacional *Microsoft Windows 7*. Este software foi desenvolvido a partir deste trabalho e não deverá ser utilizado para fins comerciais. Esta informação é exibida inclusive na tela inicial do aplicativo e também no menu Ajuda > Sobre (tecla de atalho F1).

O aplicativo foi limitado para solução apenas de vigas, descartando então a possibilidade de inserção de outros elementos estruturais como treliças e pórticos por exemplo. Todavia, ele foi programado utilizando o método da rigidez direta para solucionar inclusive vigas hiperestáticas.

## **2.1. CONCILIAÇÃO ENTRE COMPUTAÇÃO E ENGENHARIA**

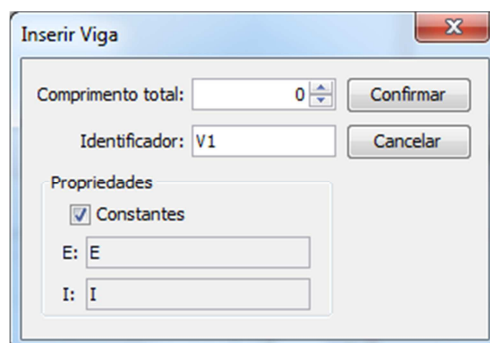
Conhecida a base teórica, iniciou-se a conciliação dos conceitos na programação. Considerando que o *Java* suporta a programação orientada a objetos, todo o aplicativo foi desenvolvido utilizando esse método que facilita demais na definição de características específicas de cada elemento.

### **2.1.1. INSERIR UMA VIGA**

Como esse aplicativo tem o propósito educativo, a tentativa foi de criar uma interface bastante simples para o usuário. Um dos primeiros passos para a resolução do método da rigidez direta é a definição de todos os nós e membros que compõem a estrutura. A localização de uma carga pontual é considerada um nó na viga, assim como um apoio e um momento. Então, para simplificar a configuração da viga, o aplicativo realiza a definição dos nós e membros automaticamente. Com isso, o usuário precisa apenas informar o comprimento total da viga, incluindo balanços e

depois continuar a configuração dos apoios e cargas. Por exemplo, ao inserir uma carga distribuída, os nós da localização inicial e final são criados automaticamente. Outra característica importante do método da rigidez direta é a utilização do módulo de elasticidade e do momento de inércia da seção de cada membro. Embora no mundo acadêmico em geral, grande parte das situações criadas na teoria são vigas de material e seção constantes em todos os membros. Em alguns casos, essas características nem mesmo são consideradas, pois o objetivo é apenas encontrar as reações nos apoios e construir os diagramas. Além disso, o principal objetivo deste método de cálculo é obter os deslocamentos nodais, mas demonstrar os deslocamentos acaba fugindo dos objetivos deste trabalho. Portanto, o aplicativo torna opcional a utilização do módulo de elasticidade e momento de inércia da viga, podendo ser consideradas constantes, conforme demonstrado na Figura 1.

Figura 1: Tela de inserção de vigas



Fonte: Do Autor, 2015.

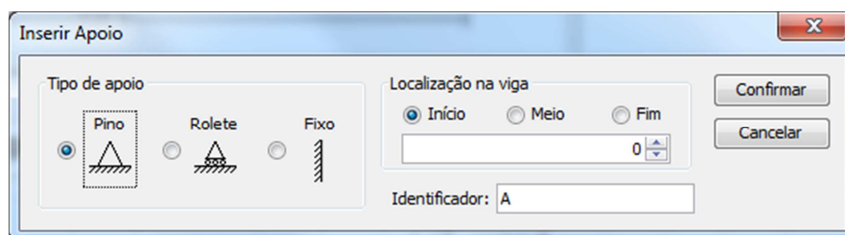
Além da viga, o aplicativo permite a inserção dos elementos: apoio, carga pontual, carga distribuída e momento. Todas essas opções, incluindo a viga, estão disponíveis no menu Inserir do aplicativo e serão apresentadas a seguir.

### 2.1.2. INSERIR APOIOS

A primeira restrição para inserção de um apoio é que o aplicativo só permite isso se a viga já tiver sido inserida. Isto evita do usuário utilizar uma localização fora do comprimento da viga, localização esta que é referente ao início da viga, da esquerda para a direita.

Os tipos de apoio disponíveis são rolete, pino e fixo que são apoios do primeiro, segundo e terceiro gênero respectivamente. No caso do apoio fixo, conhecido também como engastado, a sua localização está restrita ao início ou fim da viga. A Figura 2 demonstra a tela de inserção dos apoios.

Figura 2: Tela de inserção de apoio



Fonte: Do Autor, 2015.

### 2.1.3. INSERIR CARGAS

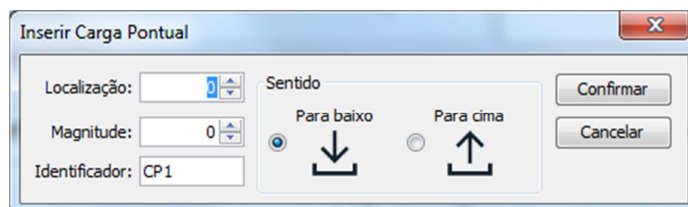
Os carregamentos disponíveis para o usuário são cargas pontuais, distribuídas e momentos. No caso das cargas distribuídas é permitida utilização de cargas uniformes ou lineares com formato triangular ou trapezoidal. Uma carga só é permitida ser adicionada quando o usuário inseriu a viga e ao menos um apoio. A sua localização segue as mesmas condições de localização dos apoios. O único diferencial está nas cargas distribuídas que possuem uma localização inicial e final. Portanto foi adicionado mais uma restrição à localização final que deve obrigatoriamente ser maior do que a inicial.

A convenção de sinais utilizada é a seguinte: cargas pontuais e distribuídas com sentido para cima tem valor positivo e para baixo, negativo. Já para os momentos o sentido anti-horário tem valor positivo e horário, negativo. Embora o usuário precise apenas escolher o sentido da carga com as opções disponibilizadas, pois a alteração dos sinais é realizada automaticamente pelo aplicativo. Além disso, nas telas de inserções de cargas, foi criada a restrição para utilização apenas de carregamentos verticais, no eixo Y e momentos em torno do eixo Z. Isso propositalmente limita a viga a possuir apenas esforço cortante e momento fletor, que são as situações mais comuns estudadas para esse tipo de estrutura.



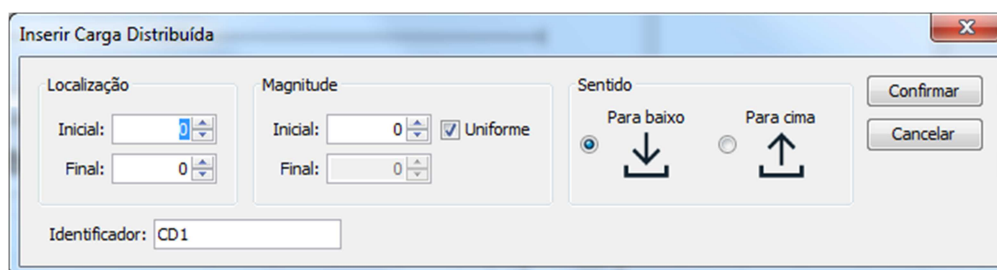
As Figuras 3, 4 e 5 demonstram as telas de inserção de cada uma das cargas disponíveis, onde é possível inclusive perceber as opções de sentido.

Figura 3: Tela de inserção de carga pontual



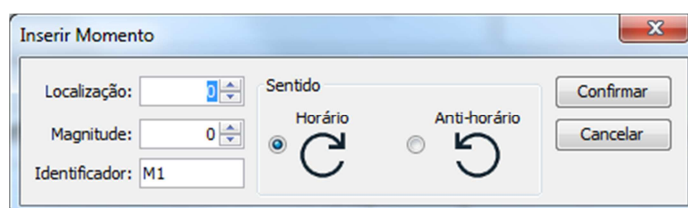
Fonte: Do Autor, 2015.

Figura 4: Tela de inserção de carga distribuída



Fonte: Do Autor, 2015.

Figura 5: Tela de inserção de momento



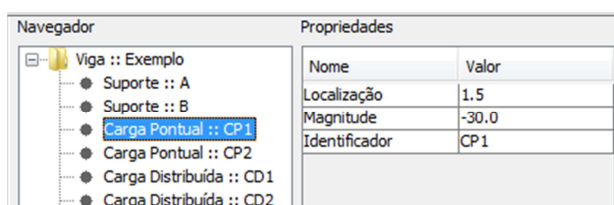
Fonte: Do Autor, 2015.

#### 2.1.4. PAINEL DE NAVEGAÇÃO E PROPRIEDADES

Cada elemento inserido no aplicativo é demonstrado em um navegador no painel direito. Ele será utilizado para modificar ou remover os elementos adicionados. Ao lado do navegador, existe um quadro que exhibe as propriedades de cada elemento selecionado. Juntos eles formam um painel que deve tornar a localização de cada elemento mais prática. Para modificar um elemento, é necessário um duplo clique no item desejado no navegador ou pressionar a tecla

*Enter* do teclado quando o item estiver selecionado. Para remover, basta pressionar a tecla *Delete* do teclado sobre o item selecionado. Quando o elemento a ser removido for a viga, o aplicativo irá remover todos os demais elementos, já que dependem da viga para existirem. Todavia, antes da remoção de qualquer elemento, é emitido uma mensagem para confirmar a exclusão, evitando assim uma remoção acidental. A Figura 6 mostra o painel de navegação do aplicativo com alguns elementos adicionados:

Figura 6: Painel de navegação



Fonte: Do Autor, 2015.

### 2.1.5. CÁLCULO

A ideia do aplicativo é que ele realizasse os cálculos em tempo real, por isso não existe um botão ou ação para calcular ou solucionar a viga. O cálculo é feito automaticamente após cada interação com o usuário. Isto é, a cada inserção, modificação ou remoção realizada, o cálculo é efetuado. Com isso, um dos grandes desafios de apresentar os diagramas em tempo real é que, dependendo da configuração que o usuário realizar, a viga pode não ter uma solução válida. Para evitar erros nos cálculos ou até mesmo um erro de execução no próprio aplicativo, por situações desconhecidas provocadas pelo usuário, foi criada uma variável interna para controlar a estabilidade da viga. De forma resumida essa variável é iniciada considerando a viga instável. Se todos os cálculos do método da rigidez direta forem realizados corretamente a viga passa a ser considerada estável, demonstrando então os resultados em tela. Portanto enquanto o aplicativo aguarda a configuração de uma viga estável, os diagramas estarão com a mensagem: "Sem dados calculados". Se quando o usuário terminar a configuração da viga, não for encontrada uma solução para a mesma, essa viga será definida como instável e nos diagramas será apresentada a mensagem: "Viga instável. Por favor, verifique as



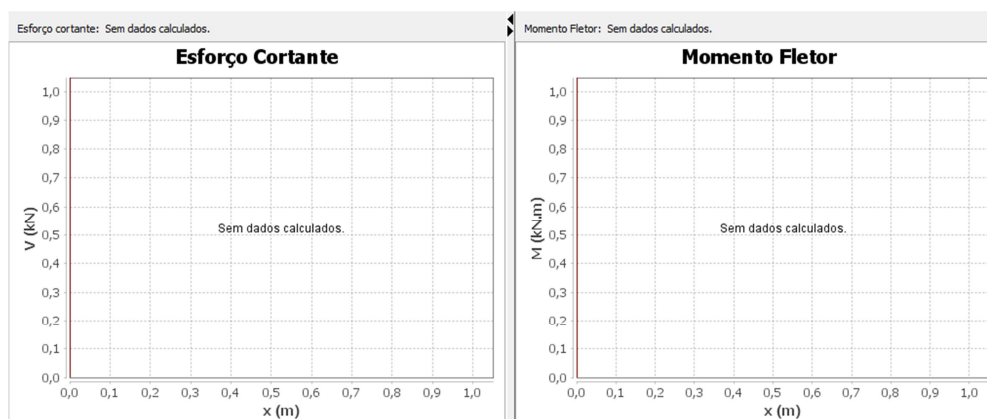
condições de apoio". Essa mensagem é apenas uma sugestão, já que a causa mais provável da instabilidade da viga seja a configuração incorreta dos apoios, resultando em uma viga hipostática.

## 2.1.6. PAINEL DE DIAGRAMAS

Na parte inferior do aplicativo está localizado o painel com os diagramas de esforço cortante e momento fletor. Como foi explicado anteriormente, o *software* está restrito para cálculos apenas de vigas e com cargas verticais e momentos em torno do eixo Z. Portanto não há necessidade do diagrama de esforço normal, pois as restrições foram criadas propositalmente para que esse esforço não existisse.

Para demonstrar os diagramas foi utilizada a biblioteca gráfica aberta *JFreeChart* que possui todos os recursos necessários para construção de gráficos. Para ambos os diagramas foi utilizado o gráfico do tipo linha, criando primeiramente um gráfico para demonstrar a viga e seus respectivos nós e depois um gráfico com o diagrama propriamente dito. Além disso, foi criado um algoritmo para demonstrar os valores em cada nó bem como os pontos máximos e mínimos de cada trecho. A Figura 7 demonstra os gráficos vazios, aguardando a configuração de uma viga, com a mensagem "Sem dados calculados", conforme explicado anteriormente.

Figura 7: Painel de diagramas



Fonte: Do Autor, 2015.

Ao clicar em qualquer trecho do gráfico, acima do diagrama é possível visualizar o valor correspondente para o ponto x escolhido, conforme é demonstrado na Figura 8, onde uma viga foi previamente configurada.

Figura 8: Esforço cortante no ponto x

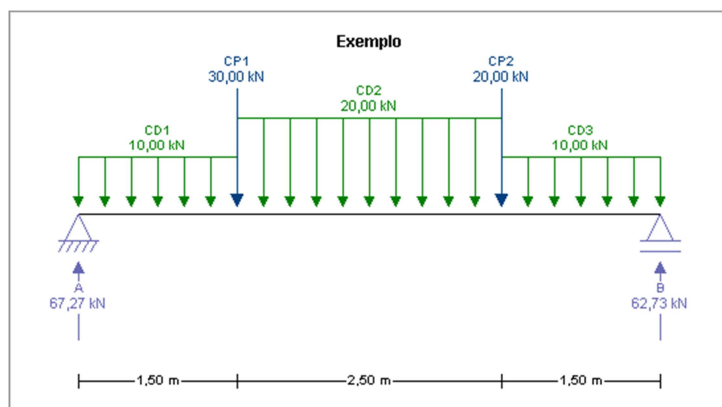


Fonte: Do Autor, 2015.

### 2.1.7. PAINEL DE VISUALIZAÇÃO DA VIGA

O painel superior esquerdo demonstra de forma simples uma visualização da configuração da viga em duas dimensões. Todos os elementos podem ser mostrados/ocultados através do menu Exibir. Portanto, caso o usuário não deseje visualizar as reações nos apoios por exemplo, poderá desabilitar acessando o menu Exibir > Reações. A Figura 9 demonstra uma viga configurada no painel de visualização:

Figura 9: Visualização da configuração da viga



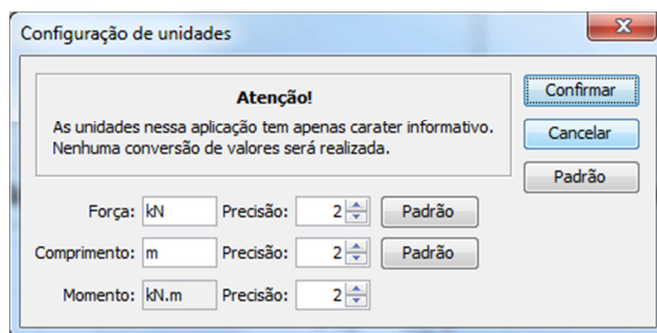
Fonte: Do Autor, 2015.

### 2.1.8. UNIDADES

As unidades padrão de configuração são metro (m) para comprimentos e quilo Newton (kN) para força. A unidade do momento é definida automaticamente de acordo com o comprimento e força, nesse caso por padrão é quilo Newton vezes metro (kN.m).

O método da rigidez direta traz os resultados na unidade de medida que foi utilizada inicialmente, ou seja, não há conversão de unidades durante a resolução. Portanto, nesse aplicativo as unidades servem apenas para facilitar a identificação dos valores na visualização da viga e nos diagramas. O que significa que nenhuma conversão de valores é realizada quando o usuário altera as unidades. Por exemplo, se uma carga de 10 kN é inserida, ao trocar a unidade das forças de kN para N, a carga simplesmente será demonstrada como 10 N. Essa informação foi deixada também na tela de unidades do aplicativo, conforme pode ser observado na Figura 10.

Figura 10: Tela de configuração de unidades



Fonte: Do Autor, 2015.

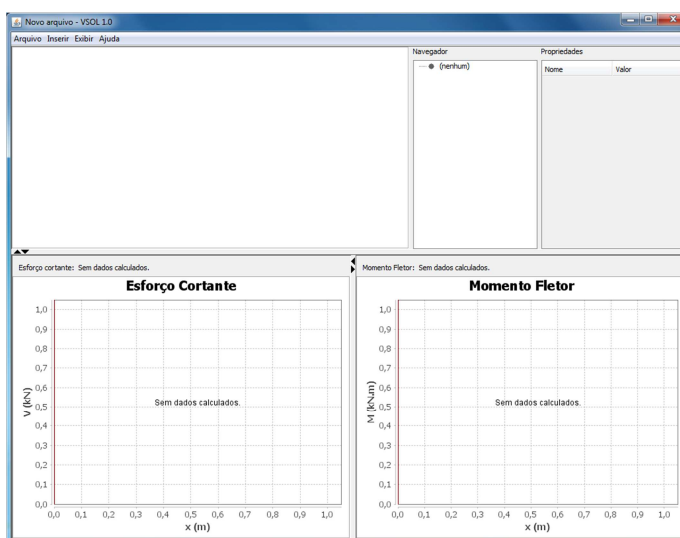
### 2.1.9. VSOL

Com o intuito de facilitar a divulgação e até mesmo a sua localização e aparência no sistema operacional, foi pensado em um nome para o aplicativo, algo que fosse sucinto e lembrasse solução e simulação de vigas. Após algumas tentativas e pesquisas o nome ficou definido como VSOL.

No aplicativo foi implementado o controle de arquivos, portanto existe o menu Arquivo, com as opções Novo, Abrir, Salvar e Salvar como. Os arquivos criados pelo aplicativo têm a extensão ".vsol" no sistema operacional.

A Figura 11 mostra a imagem da tela principal do aplicativo com todos os painéis explicados anteriormente:

Figura 11: Tela principal

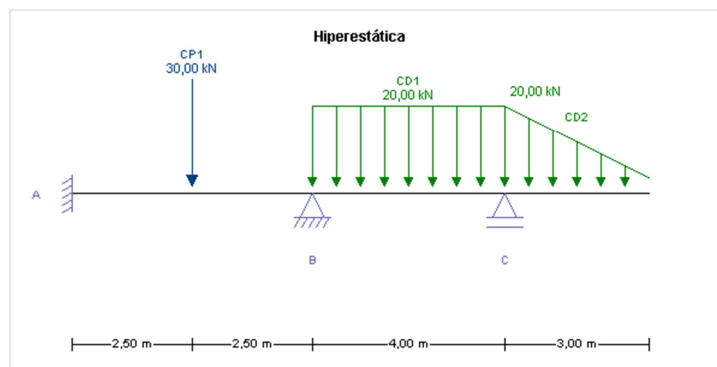


Fonte: Do Autor, 2015.

### 3. RESULTADOS E DISCUSSÕES

Durante o desenvolvimento do aplicativo, vários exemplos utilizados em livros e em sala de aula foram testados para confirmar os resultados obtidos. Neste trabalho, contudo, será apresentada uma viga hiperestática, contemplando os diferentes tipos de apoios e carregamentos oferecidos pelo aplicativo. Os resultados foram testados manualmente, contudo para simplificar a apresentação, eles serão comparados com o *software* aberto FTOOL, amplamente utilizado no meio acadêmico para resolução de estruturas em duas dimensões. A viga teórica criada possui a configuração apresentada na Figura 12.

Figura 12: Viga hiperestática configurada no VSOL

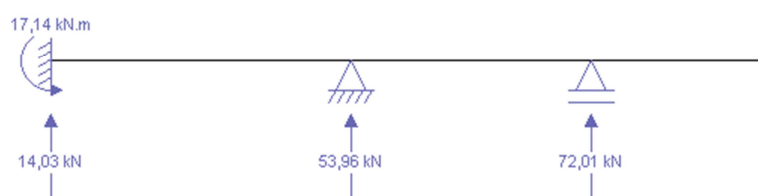


Fonte: Do Autor, 2015.

### 3.1. REAÇÕES NOS APOIOS

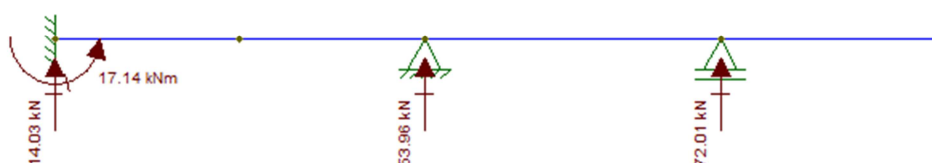
Conforme se pode observar nas Figuras 13 e 14, as reações obtidas nos apoios em ambos os *softwares* são idênticas, sendo elas um momento de 17,14 kN.m e uma força de 14,03 kN no apoio A, uma força de 53,96 kN no apoio B e por fim uma força de 72,01 kN no apoio C.

Figura 13: Reações nos apoios obtidas no VSOL



Fonte: Do Autor, 2015.

Figura 14: Reações nos apoios obtidas no FTOOL

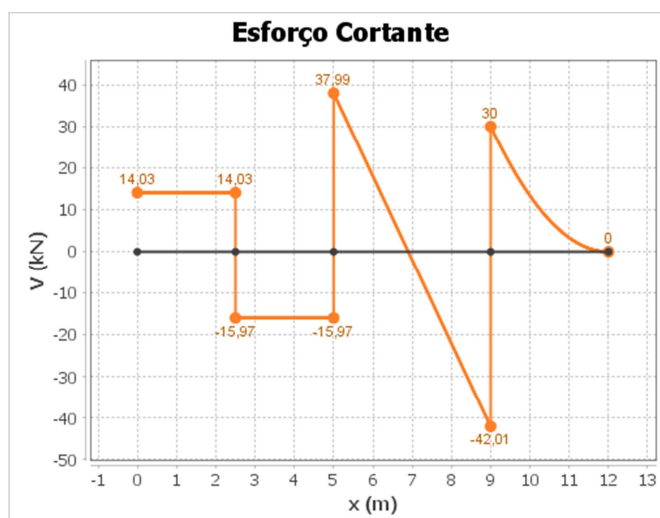


Fonte: Do Autor, 2015.

### 3.2. ESFORÇO CORTANTE

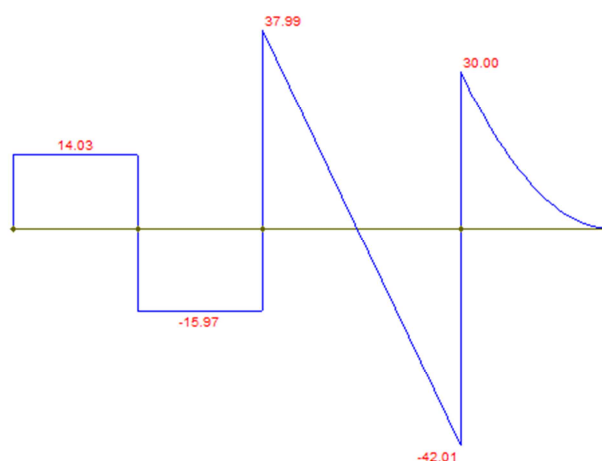
Com as reações definidas, as equações do esforço cortante para cada trecho podem ser encontradas calculando-se os esforços internos normalmente. Pode-se observar nas Figuras 15 e 16 que os gráficos, apesar de escalas diferentes, apresentam os mesmos valores. Um detalhe interessante que pode ser observado nesse diagrama é a forma do gráfico no último trecho decorrente da equação quadrática da carga triangular.

Figura 15: Esforço cortante no VSOL



Fonte: Do Autor, 2015.

Figura 16: Esforço cortante no FTOOL



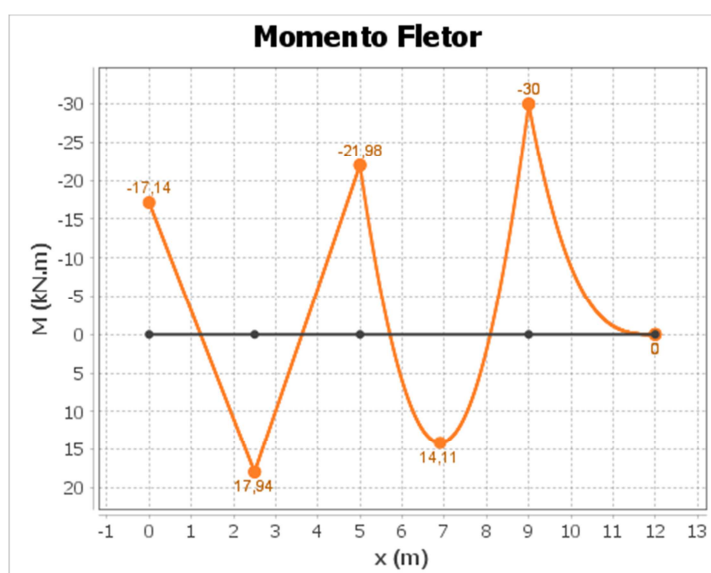
Fonte: Do Autor, 2015.



### 3.2.1. MOMENTO FLETOR

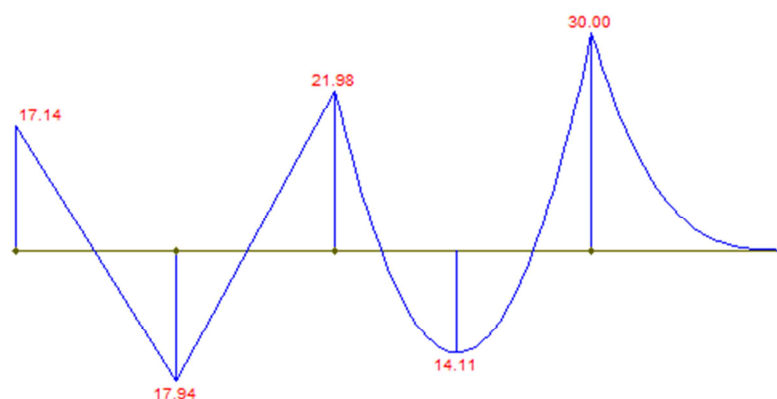
Assim como os diagramas apresentados no esforço cortante, as Figuras 17 e 18 demonstram os diagramas do momento fletor nos dois aplicativos. Pode-se observar no terceiro trecho a parábola formada pela equação quadrática do momento fletor resultante da carga uniformemente distribuída e no último trecho a forma do gráfico para a equação cúbica da carga triangular.

Figura 17: Momento fletor no VSOL



Fonte: Do Autor, 2015.

Figura 18: Momento fletor no FTOOL



Fonte: Do Autor, 2015.

#### 4. CONCLUSÕES

Além do aplicativo oferecer um número infinito de configurações de vigas que podem ser criadas, um dos grandes diferenciais é a possibilidade de visualizar a construção dos diagramas passo a passo a cada ação realizada pelo usuário. Isso amplia as possibilidades de utilização do aplicativo, inclusive na sala de aula.

Considerando as características do aplicativo e os diferenciais citados ao longo do trabalho, uma das sugestões de uso seria o professor configurar uma viga em conjunto com os alunos. A cada carga inserida poderiam observar e discutir as mudanças nas reações e nos diagramas. Outra alternativa seria o professor montar toda a configuração de uma viga, porém deixar as reações e diagramas ocultos, configurações estas que são possíveis no aplicativo. Então o professor poderia deixar os alunos resolverem a viga e depois compararem e discutirem os resultados obtidos com os exibidos no aplicativo. Por fim, outra utilização interessante seria ao final da configuração de uma viga, modificar a posição dos apoios, intensidade ou localização das cargas, para visualizar e debater as mudanças ocorridas. Por exemplo: um apoio de uma extremidade ser trocado do segundo gênero para terceiro, onde será criado um momento como reação, alterando consideravelmente o diagrama do momento fletor. Essas são apenas sugestões de como esse software pode ser versátil e auxiliar no processo de desenvolvimento das aulas de análise estrutural.

Portanto a conclusão é que o *software* atingiu os objetivos desejados. Com uma interface agradável, simples e intuitiva, não há necessidade de um manual de instruções para utilizar essa ferramenta educacional. Uma das sugestões para trabalhos futuros é realizar uma pesquisa de aproveitamento do *software*. Por exemplo: comparar o desempenho dos alunos nas avaliações de um semestre em que foi utilizado o *software* com um semestre anterior onde não foi utilizado. Essa seria uma tentativa de mensurar se realmente houve melhora na aprendizagem com a utilização dessa ferramenta. Outra sugestão seria realizar uma pesquisa com professores e acadêmicos sobre a utilidade deste aplicativo, com uma coleta de opiniões e críticas de mudanças e implementações que poderiam ser feitas para ampliar a gama de possibilidades que o aplicativo pode proporcionar. Por exemplo: suportar vigas Gerber, demonstrar os deslocamentos, linha elástica, um módulo para

calcular treliças, outro para pórticos. Enfim, se tratando de um *software*, as possibilidades são ilimitadas.

## 5. REFERÊNCIAS BIBLIOGRÁFICAS

BEER, Ferdinand P.; JOHNSTON JR., Elwood Russell. **Mecânica vetorial para engenheiros: estática**. 5. ed. São Paulo: Makron Books, 1991. 793 p.

BEER, Ferdinand Pierre; JOHNSTON JR, Elwood Russell. **Resistência dos materiais**. 2. ed. São Paulo: McGraw-Hill, 1989. 654 p.

MARTHA, Luiz Fernando. **Análise de estruturas: conceitos e métodos básicos**. Rio de Janeiro: Elsevier, 2010. 524 p.

VALENTE, José Armando et al. Diferentes usos do computador na educação. **Computadores e Conhecimento: repensando a educação**, p. 1-23, 1993. Disponível em: <http://ffalm.br/gied/site/artigos/diferentesusoscomputador.pdf>

VALENTE, José Armando. **O computador na sociedade do conhecimento**. Campinas, SP: UNICAMP/NIED, 1999.

ARNOLD, Ken; GOSLING, James. **A linguagem de programação Java**. 4 ed. Porto Alegre: Bookman, 2007.

LACERDA, Rafael de Alencar. **Proposta de um modelo para análise de requisitos de software educativo**. 2007. 114 f. Dissertação (Mestrado em Educação) - Universidade de Brasília, Brasília.

**FTOOL: Um Programa Gráfico-Interativo para Ensino de Comportamento de Estruturas**: Luiz Fernando Marha, Porto Alegre - RS, PUC-Rio, 2012. v.3.00. Disponível em: <http://www.tecgraf.puc-rio.br/ftool/>. Acesso em: 26/06/2014.